MRO-PUF: Physically Unclonable Function With Enhanced Resistance Against Machine Learning Attacks Utilizing Instantaneous Output of Ring Oscillator

AUTHOR(S):

HIROMOTO, Masayuki; YOSHINAGA, Motoki; SATO, Takashi

CITATION:

HIROMOTO, Masayuki ...[et al]. MRO-PUF: Physically Unclonable Function With Enhanced Resistance Against Machine Learning Attacks Utilizing Instantaneous Output of Ring Oscillator. IEICE Transactions of Fundamentals on Electronics, Communications and Computer Sciences 2018, E101-A(7): 1035-1044

ISSUE DATE:

2018-07-01

URL:

http://hdl.handle.net/2433/242228

PAPER *Special Section on Design Methodologies for System on a Chip*

MRO-PUF: Physically Unclonable Function with Enhanced Resistance against Machine Learning Attacks Utilizing Instantaneous Output of Ring Oscillator

Masayuki HIROMOTO[†a)], ***Member***, **Motoki YOSHINAGA**[†b)], ***Nonmember***, ***and*** **Takashi SATO**[†c)], ***Member***

SUMMARY This paper proposes MRO-PUF, a new architecture for ring-oscillator-based physically unclonable functions (PUFs) with enhanced resistance against machine learning attacks. In the proposed PUF, an instantaneous output value of a ring oscillator is used as a response, whereas the most existing PUFs directly use propagation delays to determine the response. Since the response of the MRO-PUF is non-linear and discontinuous as the delay of the ring oscillator increases, the prediction of the response by machine learning attacks is difficult. Through the performance evaluation of the MRO-PUF with simulations, it achieves 15 times stronger resistance against machine learning attacks using a support vector machine compared to the existing ones such as an arbiter PUF and a bistable ring PUF. The MRO-PUF also achieves a sufficient level of the basic performance of PUFs in terms of uniqueness and robustness.

key words: *physically unclonable function (PUF), chip identification, machine learning attacks, ring oscillator*

1. Introduction

Increasing number of counterfeit chips are getting circulated in the silicon device market. One of the countermeasures for such counterfeited chips is to construct an authentication system using physically unclonable functions (PUFs) [1], which examines whether the chips are registered products or not. A PUF is a circuit that is embedded in a product chip and serves as a function $r = f_\alpha(\boldsymbol{c})$, where $\boldsymbol{c}$ is an input of the function called "challenge" and r is an output of the function called "response." The set of the challenge response pairs (CRPs), $(\boldsymbol{c}, r)$, depends on physical variation of the chips, α, making it possible for PUFs to generate chip-intrinsic responses. Figure 1 shows an example of a PUF-based chip authentication system. The manufacturer prepares a database of CRPs in advance. The chip authentication is realized by checking whether the response of the product matches the true response stored in the database or not. As long as the CRP space is large enough and the CRPs are not reused, this authentication system is safe against attackers who intercept CRPs in past transactions.

Recently, PUFs must counter a new type of attacks, namely, the machine learning attacks [2]–[4], where the attackers try to pass through the authentication process by inferring a whole CRP space of a PUF from a small subset of CRPs by using machine learning techniques. Unfortunately, most existing PUFs can be easily modeled using simple machine learning algorithms such as linear classifiers [2], [3]. This weakness owes to their circuit constructions; the response is determined by simple operations such as sum of the propagation delays. For example, an arbiter PUF [5], which consists of cascaded multiplexers to provide multiple paths with different delays depending on the challenges, determines the response just by comparing delays of the two paths. Since the propagation delay can be modeled as a simple sum of the gate delays, the responses of these PUFs that directly use delays to generate responses can be easily predicted by simple linear classifiers. A practical PUF must utilize more complex mechanism for the response determination so that its CRP space cannot be linearly separated.

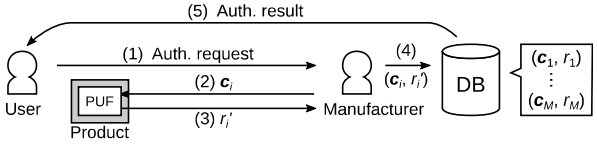

Fig. 1 An example chip authentication system using a PUF. First, a user requests an authentication process (1), then a manufacturer sends a challenge vector $\boldsymbol{c}_i$ to the PUF in the product (2) to get its response r_i' (3) and checks if the response r_i' matches the true response r_i stored in the database (4), and finally sends the result back to the user (5).

Our objective is to propose a new PUF architecture that contributes to better resistance against machine learning attacks. We propose a modulo ring oscillator PUF (MRO-PUF), whose response is determined by an *instantaneous* output of a ring oscillator (RO) instead of the path delay itself. "Instantaneous" means that the output of the RO is latched at a predefined timing to generate a response of the PUF. The output of the proposed PUF takes 0 and 1 alternatively, as the delay of the RO increases. This nonlinearity and discontinuity of the response makes the CRP space inseparable by a simple linear function, making its CRPs hard to model and predict by machine learning techniques.

The key contribution of this work is summarized as follows:

- A new PUF architecture whose response is determined by an instantaneous output of a RO is proposed.
- An analytical proof of the better resistance against ma-

Manuscript received September 11, 2017.
Manuscript revised January 16, 2018.
[†]The authors are with Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: hiromoto@i.kyoto-u.ac.jp
b) E-mail: paper@easter.kuee.kyoto-u.ac.jp
c) E-mail: takashi@i.kyoto-u.ac.jp
DOI: 10.1587/transfun.E101.A.1035

chine learning attacks of the MRO-PUF than that of the existing PUFs is presented.
- The performance of the MRO-PUF is evaluated through intensive circuit simulations. Over the existing PUFs, the MRO-PUF achieves 15 times stronger resistance against machine learning attacks, in addition to the basic performance improvement in terms of uniqueness and robustness.

The remainder of the paper is organized as follows: First, the existing PUFs and the machine learning attacks to such PUFs are reviewed in Sect. 2. Second, the proposed MRO-PUF is described and its resistance against machine learning attacks is discussed in Sect. 3. Then the performance evaluation of the PUFs is described in Sect. 4. Finally, related works are summarized in Sect. 5 and the paper is concluded in Sect. 6.

2. PUFs and Machine Learning Attacks

2.1 PUFs for Chip Authentication

There are two types of PUFs used for chip authentication systems: strong PUFs and weak PUFs [6]. With a strong PUF, the chip authentication system directly uses its CRPs as shown in Fig. 1. The security level of such authentication systems greatly depends on the complexity of the PUF's function and how large the CRP space is, i.e., the PUF itself must be "strong." On the other hand, a weak PUF is used in combination with general encryption algorithms to generate encryption keys [6], [7]. In such a system, the weak PUF does not need to be "strong" since the security level of the authentication system is ensured by the encryption algorithms. However, it is known that the encryption circuit may also be threatened by so-called side-channel attacks [8], [9].

In this work, we focus on enhancing the security performance of the strong PUFs. The strong PUFs are required to provide us with a large number of CRPs through a sufficiently complex function. Among the existing PUFs [5], [10]–[12], an arbiter PUF [5] and a bistable ring PUF [12] have advantages in the first requirement as they can take $O(2^n)$ challenges when a circuit is composed of n unit-components.

2.1.1 Arbiter PUF

The arbiter PUF [5] is a PUF architecture that determines its response by comparing signal propagation delays of two different paths selected by challenge signals. An example structure of an n-bit arbiter PUF is shown in Fig. 2. The arbiter PUF consists of a series of 2-in-2-out switch components and an arbiter circuit located at the final stage. At each switch component, the signals propagate straight or crossed depending on the corresponding challenge signal c_i. The arbiter circuit compares the arrival times of two signals, and output a response value $r = \{0, 1\}$ depending on the early arrival of upper or lower signals.

The characteristic of the arbiter PUF is formulated by

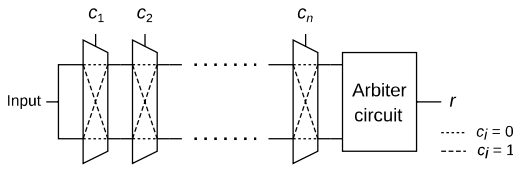

Fig. 2 A circuit structure of an arbiter PUF.

using a delay time of each switch component [2]. Let the delay difference of the two paths be $\Delta d_i(c_i)$ for the i-th switch component with a challenge signal c_i. Each $\Delta d_i(c_i)$ takes a chip-intrinsic random value because it is determined by delay variation of the transistors that compose the switch. The delay difference of the two signals arriving at the input of the arbiter circuit is

$$\Delta_{\mathrm{arb}} = \sum_{i=1}^{n} b_i \Delta d_i(c_i), \tag{1}$$

where b_i indicates whether the outputs of the i-th switch component arrive at the arbiter circuit straight ($b_i = 1$) or crossed ($b_i = -1$), which can be written as

$$b_i = \prod_{k=i}^{n} (1 - 2c_k). \tag{2}$$

The final response of the arbiter PUF is determined as

$$r = U(\Delta_{\mathrm{arb}}), \tag{3}$$

where $U(\cdot)$ is a unit step function,

$$U(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases}. \tag{4}$$

2.1.2 Bistable Ring PUF

A bistable ring PUF (BR-PUF) [12] is another PUF architecture that has multiple selectable paths. The BR-PUF consists of bistable rings (BRs), which are inverter rings with even number of stages having two stable states. Figure 3 shows an example four-stage BR, which can take one of two stable states, "0101" (Fig. 3(a)) or "1010" (Fig. 3(b)). If an unstable initial state, such as "0000" or "1111", is given, the state of the BR will converge to either "0101" or "1010" after a certain period of time. The final state is determined by delay variations of the inverters in the BRs, hence a logic state of a node in the circuit can be used as a response.

The BR-PUF employs multiple route-selectable stages to attain an exponential number of challenge-response combinations. Figure 4 shows a circuit structure of an n-bit BR-PUF [12]. Each stage consists of two NOR gates in between a demultiplexer and a multiplexer, which operates synchronously to select one of the NOR gates by a given challenge signal c_i. The NOR gates behave as inverters in Fig. 3 when the RESET signal is deasserted. The procedure to obtain a response of the BR-PUF is, first setting

(a) '0101" state (b) '1010" state

Fig. 3 Two stable states of a four-stage bistable ring.

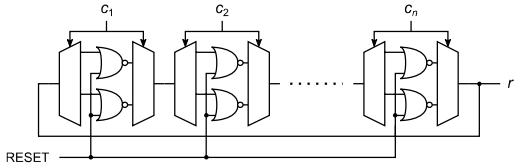

Fig. 4 A circuit structure of a bistable ring PUF.

RESET=1 to initialize the ring state and assigning challenges $c_1, c_2, \cdots, c_n$ to select corresponding NOR gates, and then setting RESET=0 to start oscillation. After the ring state is converged, the ring state is read out as a response r.

The BR-PUF is reported to yield a complex response to a challenge due to its time evolving behavior [12]. In a recent work, an approximation model for the BR-PUF to generate response has been proposed [13]. In this model, the difference of the "driving force" between the odd- and the even-stage transistors is considered. For example, in the case of Fig. 3, if the driving force of the nMOS transistors in the odd stages and the pMOS in the even is strong, the ring is more likely to be "0101" state than "1010." For the i-th NOR gate selected by a challenge c_i, let us define $f_i + c_i \Delta f_i$ as difference of the driving force of the pMOS against the nMOS. The total difference of the driving force of the BR-PUF becomes

$$\Delta_{\mathrm{BR}} = \sum_{i=1}^{n} (-1)^i (f_i + c_i \Delta f_i), \tag{5}$$

and the final response of the BR-PUF is determined as

$$r = U(\Delta_{\mathrm{BR}}). \tag{6}$$

2.2 Machine Learning Attacks

The arbiter PUF and the BR-PUF are area-efficient and suitable for use as strong PUFs since they can generate 2^n CRPs for n stages. However, they are reported to be vulnerable against the machine learning attacks [13], [14].

A machine learning attack against a PUF that returns binary response is defined as the following problem: given a partial set of CRPs $\{(\boldsymbol{c}_1, r_1), (\boldsymbol{c}_2, r_2), \cdots, (\boldsymbol{c}_m, r_m)\}$ $(m < 2^n)$ as a training set, classify an unknown challenge $\boldsymbol{c}_i$ into either of the two classes; $r_i = 0$ or $r_i = 1$. The most common approach to solve such classification problems is to apply a linear support vector machine (SVM) [15], which classifies an input vector $\boldsymbol{x}$ by

$$y = \mathrm{sign}(\boldsymbol{w}^T \boldsymbol{x}), \tag{7}$$

where $\boldsymbol{w}$ is a support vector that defines a hyper-plain to separate the two classes. The linear SVM can be extended to non-linear problems by introducing a kernel function $\phi(\boldsymbol{x})$.

The arbiter PUF is known to be modeled by the SVM classifier [14]. Equation (3) can be rewritten as

$$r = U(\Delta_{\mathrm{arb}}) = U(\boldsymbol{w}^T \boldsymbol{b}), \tag{8}$$

where the support vector $\boldsymbol{w} = (w_1, w_2, \cdots, w_{n+1})^T$ is

$$\begin{cases} w_1 = (\Delta d_1(0) + \Delta d_1(1))/2 \\ w_i = (\Delta d_{i-1}(0) - \Delta d_{i-1}(1) + \Delta d_i(0) + \Delta d_i(1))/2 \\ \qquad\qquad (i = 2, 3, \cdots, n) \\ w_{n+1} = (\Delta d_n(0) - d_n(1))/2 \end{cases} \tag{9}$$

and $\boldsymbol{b} = (b_1, b_2, .., b_n, 1)^T$ is the vector of indicators determined from challenges $\boldsymbol{c}$ as in Eq. (2). Thus, by using a kernel function $\boldsymbol{b} = \phi(\boldsymbol{c})$, this indicates that the arbiter PUF can be modeled by an SVM classifier.

The BR-PUF can also be attacked by a linear SVM classifier [13]. Equation (6) can be rewritten as

$$r = U(\Delta_{\mathrm{BR}}) = U(F + \boldsymbol{w}^T \boldsymbol{c}), \tag{10}$$

where $\boldsymbol{w} = (w_1, w_2, \cdots, w_{n+1})^T$, $w_i = (-1)^i \Delta f_i$ is a support vector and $F = \sum_{i=1}^{n} (-1)^i f_i$ is a constant value. This shows that the BR-PUF can be modeled by a linear SVM without using a kernel function.

3. Modulo Ring Oscillator PUF

As described in the previous section, the existing PUFs have vulnerability for machine learning attacks, such as by SVM classifiers. In this paper, we propose a new PUF architecture that utilizes a modulo operation, "$\lfloor x \rfloor$ mod 2", instead of the unit step function $U(x)$ to determine a response. This section describes our proposed PUF named "modulo ring oscillator PUF (MRO-PUF)," and discusses its resistance against machine learning attacks.

3.1 Basic Concept

We first explain the basic concept of the MRO-PUF using a simple RO circuit as shown in Fig. 5. In this figure, three RO instances #1–3 having slightly different oscillation periods of $T_{\mathrm{RO\#1}} < T_{\mathrm{RO\#2}} < T_{\mathrm{RO\#3}}$ are illustrated as an example. At an oscillation time of T_{meas}, due to the process variation, phase difference among the ROs is observed. This is used as a randomness to design our proposed PUF. The response generation in this PUF can be formulated using a modulo operation,

$$r = \left\lfloor \frac{T_{\mathrm{meas}}}{T_{\mathrm{RO}}/2} \right\rfloor \bmod 2. \tag{11}$$

Here, $\lfloor T_{\mathrm{meas}}/(T_{\mathrm{RO}}/2) \rfloor$ is a toggle count of the RO during the T_{meas} period. When T_{meas} is set as a constant, then

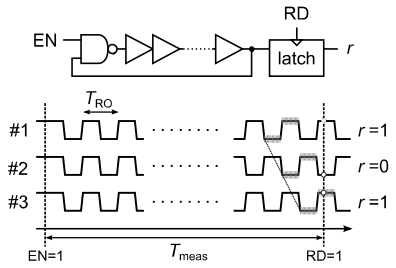

Fig. 5 Concept of the MRO-PUF. The response r alternately takes 0 or 1 depending on the RO's oscillation period T_{RO}.

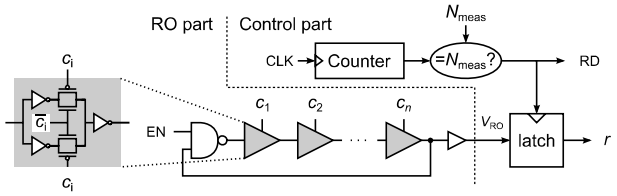

Fig. 6 A circuit structure of the proposed MRO-PUF (1-MRO-PUF). The PUF consists of a route-selectable RO, a counter, and a latch. Each buffer element in the RO contains two inverters, which are exclusively selected by a challenge signal c_i.

the response r is non-linear and discontinuous as a function of oscillation period, T_{RO}. Therefore, by employing a route-selectable structure as in the BR-PUF to vary T_{RO} with challenge inputs, we can realize a PUF that is more resistant against machine learning attacks than the existing ones.

3.2 Circuit Structure

An example circuit realization of the proposed n-bit MRO-PUF is shown in Fig. 6. The MRO-PUF consists of three major components: a route-selectable RO, a counter, and a latch. The inputs are challenge signals $c_1, c_2, \cdots, c_n$, an enable signal EN, a reference number of oscillation counts N_{meas}, and an external clock signal CLK. The outputs are a response r and a read-out signal RD.

Each buffer element in the RO (shaded triangles in Fig. 6) contains two parallel inverters. One of the two inverters is exclusively selected by a pair of pass transistors controlled by a challenge signal c_i. With this structure, the RO's oscillation period T_{RO} can be changed according to a set of challenge signals $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)^T$. The T_{RO} for a challenge vector $\boldsymbol{c}$ can be expressed as

$$T_{RO}(\boldsymbol{c}) = 2\sum_{i=1}^{n}(d_i + c_i\Delta d_i), \tag{12}$$

where d_i and $d_i + \Delta d_i$ are the delays of the i-th buffer element when $c_i = 0$ and $c_i = 1$, respectively. The enable signal EN is used to start and stop oscillation of the RO.

The counter measures the oscillation time T_{meas} by counting up the number of the external clock (CLK) pulses. When the count reaches the predefined reference number N_{meas}, i.e., at the time of $T_{meas} = N_{meas}T_{CLK}$, the read-out signal RD is asserted to latch the output of the RO. Figure 7

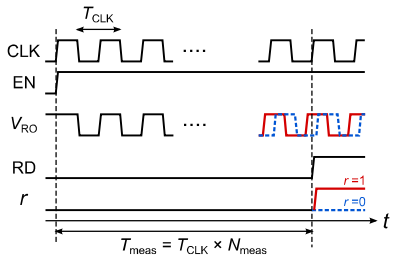

Fig. 7 A timing chart of response generation by MRO-PUF. First, the enable signal EN is asserted at a rising edge of the external clock CLK, and then, after counting up the number of clock pulses to N_{meas}, RD is asserted to latch the RO's output V_{RO} as a response r.

shows a timing diagram of the operation of MRO-PUF.

From Eqs. (11) and (12), the response of the MRO-PUF is formulated as

$$r = \left\lfloor \frac{T_{meas}}{\sum_{i=1}^{n}(d_i + c_i\Delta d_i)} \right\rfloor \bmod 2. \tag{13}$$

Since the parameters d_i and Δd_i depend on the variability of the transistors that compose the buffer element, a unique set of CRPs is obtained for each chip.

3.3 Resistance for Machine Learning Attacks

The CRP set of the MRO-PUF is difficult to linearly separate. The response determination shown in Eq. (13) can be rewritten as

$$r = \left\lfloor \frac{T_{meas}}{D + \boldsymbol{w}^T\boldsymbol{c}} \right\rfloor \bmod 2, \tag{14}$$

where $\boldsymbol{w} = (\Delta d_1, \Delta d_2, .., \Delta d_n)^T$, and $D = \sum_{i=1}^{n} d_i$ is a constant value. In general, in order for a CRP set of a PUF to be linearly separable, a response r can be expressed as a monotonic function of a linear combination of the challenges $\boldsymbol{c}$. In Eq. (14), although the denominator of the fraction is a linear combination of the challenges, the response is determined by a modulo operation, which can even become a non-monotonic function when T_{meas} is sufficiently large as compared to the oscillation period variation. As a result, CRP of the MRO-PUF is inseparable with a linear function.

In addition, the MRO-PUF has resistance against SVM attacks even when a kernel function $\phi(\cdot)$ is used. Here we consider an approximation of the modulo operation "$\lfloor\cdot\rfloor \bmod 2$" by using a unit step function $U(\cdot)$,

$$\lfloor x \rfloor \bmod 2 \simeq U(\sin(\pi x)). \tag{15}$$

Although the boundary values when x is integers are not strictly correct, this gives a close approximation of the modulo operation. With this, the response in Eq. (14) can be expressed using a unit step function $U(\cdot)$ as

$$r = U(\Delta_{prop}) = U\left(\sin\left(\frac{\pi T_{meas}}{D + \boldsymbol{w}^T\boldsymbol{c}}\right)\right). \tag{16}$$

The CRPs of the MRO-PUF are linearly separable via a

kernel function $\phi(\cdot)$ if the Δ_{prop} is represented by

$$\Delta_{\text{prop}} = \boldsymbol{w}'^{T}\phi(\boldsymbol{c}), \tag{17}$$

where $\boldsymbol{w}'$ only depends on the delay parameters $d_1, d_2, \cdots, d_n, \Delta d_1, \Delta d_2, \cdots, \Delta d_n$, and a kernel function $\phi(\boldsymbol{c})$ only depends on the challenges $c_1, c_2, \cdots, c_n$. However, in order to realize this representation, it is needed for the sine function to be decomposed into a polynomial function with a finite number of terms, which is very difficult if the values of $\pi T_{\text{meas}}/(D + \boldsymbol{w}^T\boldsymbol{c})$ take a sufficiently wider rage than the period of the sine function. This shows that the CRPs of our MRO-PUF are difficult to linearly separate even if a specific kernel function is used.

3.4 MRO-PUF with Improved Robustness

While the proposed MRO-PUF in Fig. 6 has a good resistance against machine learning attacks, the robustness metric may not be sufficient. As shown in Eq. (13), the response of the MRO-PUF is determined by the delay parameters d_i and Δd_i, which are affected not only by static process variation but also by dynamic fluctuation of the temperature and/or the supply voltage. This means that the response r possibly takes an opposite binary value depending on the operating environment even if the same challenge bits are given.

In order to improve robustness, we propose to use two ring oscillators as a pair as shown in Fig. 8. Hereafter, we call the MRO-PUF in Fig. 8 as "2-MRO-PUF," and the original one in Fig. 6 as "1-MRO-PUF." The 2-MRO-PUF employs an additional RO that generates the counter clock CLK, eliminating the external clock in 1-MRO-PUF. The use of the RO improves the robustness of the proposed PUF, as it was effective in the case of ring oscillator PUF (RO PUF) [11], in which the response is determined by a difference of the oscillation frequencies of two ROs. Since temperature and the supply voltage are almost equal for the paired ROs that are closely placed, the RO PUF can successfully cancel the effect of the environmental change. Our 2-MRO-PUF thus utilizes two ROs so that the final timing to latch a response also reflects the environmental change. From Eq. (11) and the equation $T_{\text{meas}} = N_{\text{meas}}T_{\text{CLK}}$, the response generation is written as

$$r = \left\lfloor \frac{N_{\text{meas}}T_{\text{CLK}}}{T_{\text{RO}}/2} \right\rfloor \bmod 2. \tag{18}$$

In the 2-MRO-PUF, T_{CLK} changes in accordance with the change of T_{RO} against the environmental change, which enhances the robustness of the MRO-PUF.

As shown in Fig. 8, the same challenge $\boldsymbol{c}$ is given to the clock generating RO as well as the response generating RO. The oscillation period of the clock RO can be written in a similar way as Eq. (12),

$$T_{\text{CLK}}(\boldsymbol{c}) = 2\sum_{i=1}^{n}(d'_i + c_i\Delta d'_i), \tag{19}$$

where d'_i and $\Delta d'_i$ are the chip-dependent delay parameters.

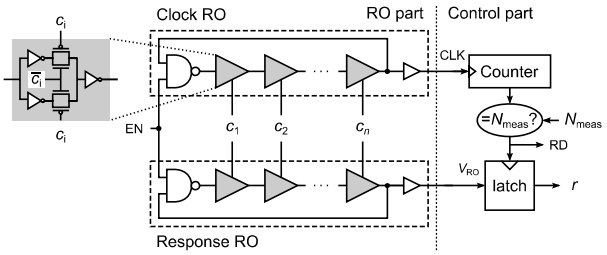

Fig. 8 A circuit structure of the 2-MRO-PUF. A clock RO is employed to generate CLK instead of using the external clock. The two ROs are both controlled by the same input challenges c_i.

The response of the 2-MRO-PUF becomes

$$r = \left\lfloor \frac{2N_{\text{meas}}\sum_{i=1}^{n}(d'_i + c_i\Delta d'_i)}{\sum_{j=1}^{n}(d_j + c_j\Delta d_j)} \right\rfloor \bmod 2. \tag{20}$$

Compared to the 1-MRO-PUF, this equation has twice the number of the delay parameters. This indicates that the 2-MRO-PUF can be more resistant against machine learning attacks than 1-MRO-PUF, in addition to enhancing the robustness.

The difficulty of the linear separation of the CRPs given by 2-MRO-PUF can be discussed in a similar way to the 1-MRO-PUF by replacing T_{meas} in Eq. (16) with

$$T_{\text{meas}} = 2N_{\text{meas}}\sum_{i=1}^{n}(d'_i + c_i\Delta d'_i). \tag{21}$$

Since the difficulty to decompose the sine function still holds in this case, the 2-MRO-PUF is also resistant against machine learning attacks, such as SVMs.

4. Evaluation

The performance of the proposed MRO-PUF is evaluated through five criteria: randomness, diffuseness, uniqueness, robustness, and resistance against machine learning attacks. Comparisons to the existing PUFs such as the arbiter PUF and the BR-PUF are also presented.

4.1 Setups

We evaluate the performance of the PUFs by simulation. When the pulse counts for the measurement, N_{meas}, is set large, long simulation time is required to obtain a response from MRO-PUF. We hence construct a simulation platform that uses both circuit-level and behavior-level simulators to accelerate the simulation speed. Figure 9 shows a simulation flow in the platform, which mainly consists of two steps: RO simulation and response generation.

In the RO simulation step, the oscillation period $T_{\text{RO}}(\boldsymbol{c}_j)$ is measured by a SPICE simulation. In our simulation, a commercial SPICE simulator and a commercial 65 nm process library are used. A set of PUF instances are generated by assuming a Gaussian distribution for threshold voltages V_{th}. For a PUF instance #i, the oscillation period $T_{\text{RO\#}i}(\boldsymbol{c}_j)$

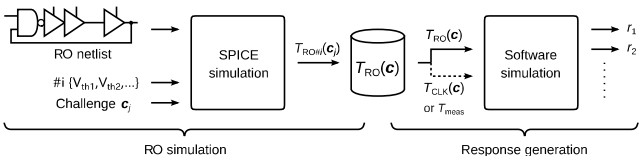

Fig. 9 A simulation flow for the MRO-PUF.

is measured and stored for all the possible challenges $\boldsymbol{c}_j$, in order to use in the succeeding software simulation. These measurements are repeated with different conditions such as temperature and supply voltage.

In the response generation step, the behavior of the controller circuit of the MRO-PUF, i.e., the counter and the response latch, is emulated by a software simulation. For a PUF instance #i, the software simulator calculates the response r_i by using $T_{\mathrm{RO\#}i}(\boldsymbol{c}_j)$ in the dataset and a given parameter T_{meas}. In the case of the 2-MRO-PUF, the clock period $T_{\mathrm{CLK\#}i}(\boldsymbol{c}_j)$ must also be stored in the dataset to calculate T_{meas} for the challenge $\boldsymbol{c}_j$.

The arbiter PUF and the BR-PUF are fully evaluated by SPICE simulations only because its simulation time is short. Similarly to the evaluation of MRO-PUF, their responses are observed under the variations of V_{th}.

4.2 Randomness, Diffuseness, and Uniqueness

First we evaluate PUF's performance with the following three metrics: randomness (H), diffuseness (D), and uniqueness (U) [16]. The randomness represents how equal the frequencies of responses 0 and 1 that a PUF returns are, the diffuseness represents whether a PUF returns different responses for different challenges, and the uniqueness represents whether different PUFs return different responses for a same challenge. Each metric takes a value between 0 and 1, where 0 is the worst and 1 is the best. We generated 10 PUF instances and gave a same set of 128 challenges to collect 1280 CRPs for 16-bit and 32-bit MRO-PUFs. Each test to collect a CRP is performed once, since the circuit simulation returns always the same results. The periods of the external clocks for 16-bit and 32-bit 1-MRO-PUFs are set to T_{CLK}=2.62 ns and T_{CLK}=5.16 ns respectively, which are the averages of the oscillation periods of the response generation ROs.

The top two graphs in Fig. 10 show the simulation results for the uniqueness of the 16-bit 1-MRO-PUF and 2-MRO-PUF. The same metrics for the arbiter PUF and the BR-PUF are compared. It is shown that all the three metrics of the MRO-PUF are improved as N_{meas} increases. With $N_{\mathrm{meas}} = 50$, for example, the proposed PUFs are comparable or superior to the existing PUFs. The bottom two graphs in Fig. 10 shows the results for the 32-bit PUFs. The 2-MRO-PUF also achieves the comparable performance of those of the 16-bit and the existing ones.

4.3 Robustness

The robustness is a characteristic that a same PUF instance always returns a same response if a same challenge is given. In this work, we evaluate the robustness with a reproduction error when the temperature and the supply voltage are changed from typical values. We generated 10 PUF instances and gave a same set of 1024 challenges to collect 10240 CRPs for 16-bit and 32-bit MRO-PUFs. The temperature is set to 30°C, 40°C, and 50°C, and the supply voltage is decreased by −1%, −5%, and −10% from the rated voltage of 0.8 V. The reproduction error is calculated against the typical setting with 25°C and 0.8 V.

Figure 11 shows the robustness of the MRO-PUFs at different operating temperatures. The reproduction error rates are plotted as a function of N_{meas}. The error rate of the 1-MRO-PUF at 30°C becomes significantly larger as N_{meas} increases. At the higher temperatures, the error rate goes up and down, which is caused by the periodic nature of the MRO-PUF. Since the error rate higher than 0.5 means that the PUF does not work correctly, the 1-MRO-PUF is unreliable under the temperature fluctuation. On the other hand, the 2-MRO-PUF achieves remarkably smaller error rate below 0.1. This shows that the utilization of the RO pair is effective to improve the robustness of the MRO-PUF.

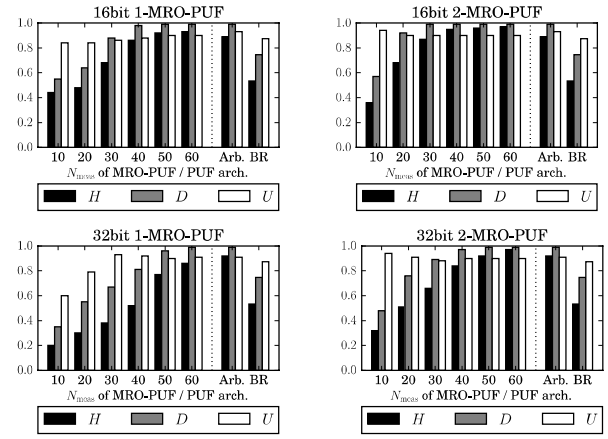

Fig. 10 Evaluation results of the PUFs' randomness, diffuseness, and uniqueness.

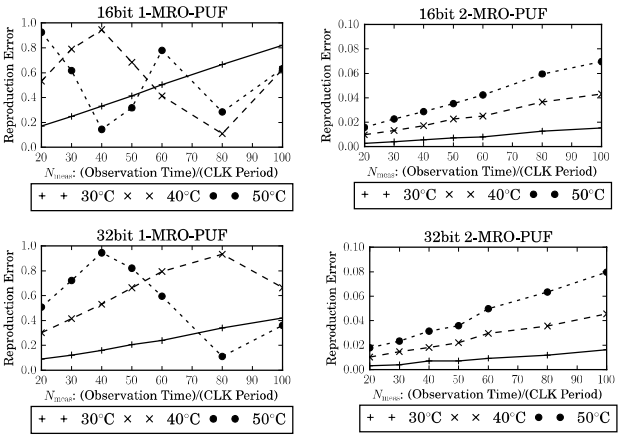

Fig. 11 Robustness of the proposed PUF under temperature variation.

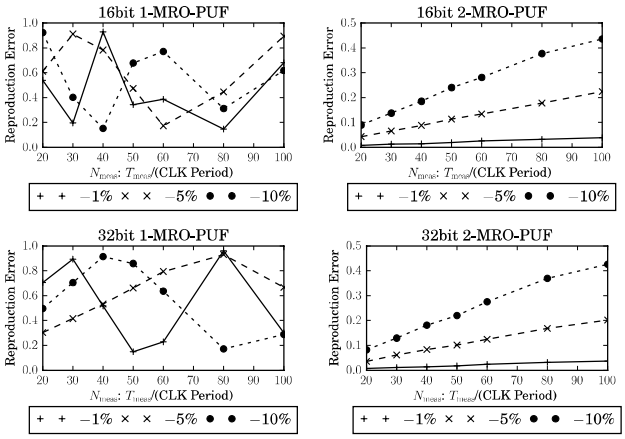

Fig. 12 Robustness of the proposed PUF under supply voltage variation.

Figure 12 shows the robustness of the MRO-PUFs at different supply voltages, in which the reproduction error rates are plotted as a function of N_{meas}. As in the case of the temperature fluctuation, the error rate of the 1-MRO-PUF is extremely large. The 2-MRO-PUF, however, achieves low error rates against the voltage fluctuation similarly to the case of temperature change. In case the error rate of the 2-MRO-PUF with −10% voltage is unacceptable, the supply voltage for the proposed PUF should be controlled within a few percent of the typical value, which is not difficult to achieve when the circuits other than PUF are made idle.

In summary, while the 1-MRO-PUF is strongly affected by the fluctuation of the temperature and the supply voltage, the improved 2-MRO-PUF realizes sufficient robustness for such environmental fluctuations.

4.4 Resistance against Machine Learning Attacks

We evaluate the resistance against machine learning attacks of the 2-MRO-PUF by using an SVM classifier. We randomly sampled n_{train} CRPs for training and 1000 CRPs for testing without overlaps. The 2-MRO-PUFs and the BR-PUF are attacked by a linear SVM without kernel functions, while the arbiter PUF is attacked using a kernel function that maps $\boldsymbol{c} \rightarrow \boldsymbol{b}$ as shown in Eq. (2) assuming the worst case. Since the modulo function in the proposed PUF is difficult to be modeled, we evaluate the performance of model-less attacks in this experiment. The SVM classifier is implemented by using Python scikit-learn library [17].

As a preliminary experiment, we examine a relationship between the robustness and the resistance against machine learning attacks of 16-bit PUFs. We have seen that the robustness of the MRO-PUF becomes worse as the parameter N_{meas} increases as shown in Sect. 4.3, while the resistance against machine learning attacks will be improved with the larger N_{meas} as discussed in Sect. 3.3. Hence, we must choose an appropriate N_{meas} value considering the trade-off between the above two metrics. In Fig. 13, the prediction accuracy of the trained SVM classifier is plotted against the reproduction rate by changing N_{meas} to be 10, 20, 30, 40, 50, 60, 80, and

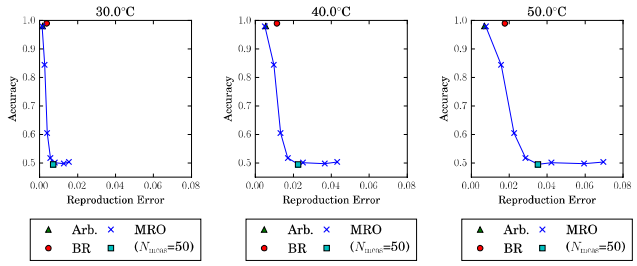

Fig. 13 Relationship between the robustness and the resistance against machine learning attacks.

100. The results of the arbiter PUF and the BR-PUF are also depicted. The accuracy of 0.5 means the highest resistance, while 0 or 1.0 means that the PUF response is completely predictable. The number of CRPs used for training is n_{train} = 500. The result shows that the 2-MRO-PUF achieves good resistance while retaining the robustness in a sufficient level. At N_{meas} = 50, marked using squares in Fig. 13, the best resistance of 0.5 accuracy is achieved. The 2-MRO-PUF is comparable to the arbiter PUF and better than the BR-PUF.

The reason why $N_{\text{meas}} \geq 50$ realizes good resistance is considered as follows. From the simulation results, the mean and the standard deviation of T_{CLK} of 16-bit 2-MRO-PUF are μ=2.62 ns and σ=0.025 ns, respectively, and those of $T_{\text{CLK}}/(T_{\text{RO}}/2)$, which appears in the model equation of Eq. (18), are μ=2.00 and σ=0.029, respectively. This means that, when N_{meas} = 50, the standard deviation of $N_{\text{meas}}T_{\text{CLK}}/(T_{\text{RO}}/2)$ becomes 1.45 and the effect of the modulo operation in Eq. (18) begins to appear. As N_{meas} increases, the above value takes the wider variation and the modulo operation works more effectively. For the 32-bit 2-MRO-PUF, we found that $N_{\text{meas}} \geq 80$ realizes good resistance based on the similar observation.

On the basis of the above results, the configuration of N_{meas} = 50 for 16-bit PUFs and N_{meas} = 80 for 32-bit PUFs is evaluated in detail. Figure 14 shows the result of the SVM attacks for 16-bit and 32-bit PUFs. The prediction accuracy by the trained SVM is plotted as a function of the number of the training CRPs, n_{train}. Since there is a variation of the PUF characteristics, the average and the best cases are plotted for the 2-MRO-PUF among the 10 instances. As shown in the results, the 2-MRO-PUF achieves lower prediction accuracy less than 0.6 whereas those of the arbiter PUF and the BR-PUF are higher than 0.9 even if a very few CRPs are used for the training. When $n_{\text{train}} = 10^4$, the prediction error for the 16-bit MRO-PUF is 45% at most whereas those of the existing PUFs are less than 3%. This means that the proposed MRO-PUF is 15 times stronger than the arbiter PUF and the BR-PUF for the SVM attacks. We can conclude that the utilization of the modulo operation as a non-linear and discontinuous function effectively works to enhance the resistance against the machine learning attacks.

Figure 14 also shows the simulation results of model-less attacks for XOR arbiter PUFs [11], which are the PUFs whose responses are determined by XOR-ing the outputs of M arbiter PUF instances. As shown in the figure, when

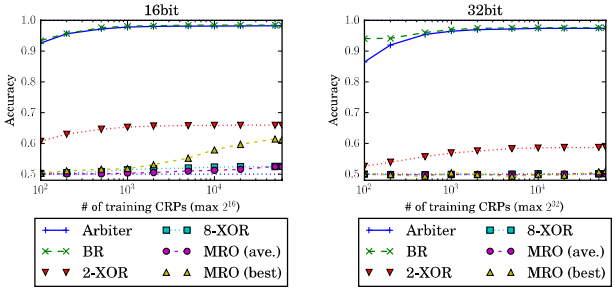

Fig. 14 Resistance against machine learning attacks by SVM classifier.

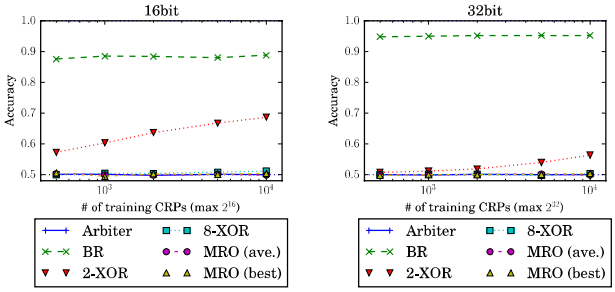

Fig. 15 Resistance against machine learning attack by evolutional strategy.

$M = 8$ (8-XOR), the XOR arbiter PUF achieves almost perfect prediction accuracy of around 0.5 and outperforms the proposed MOR-PUF. However, it is known that the XOR arbiter PUF can be predicted with 99% accuracy by modeling attacks [18]. Since there is currently no known model for the modulo function used in the MRO-PUF, the MRO-PUF still has an advantage to the XOR arbiter PUFs.

In addition to the linear classifier, we also evaluated the resistance against more advanced non-linear machine learning techniques, which are known to be effective for model-less attacks against PUFs. Figures 15–17 show the results of attacks by an evolution strategy (ES) [19], a random forest (RF) [20], and a boosting [21], respectively. The above classifiers are implemented by Python with PyBrain [22] for ES, and scikit-learn [17] for RF and boosting. For the ensemble learning methods such as the RF and the boosting, the 16-bit MRO-PUF can be modeled when a large number of training samples are used. However, the 32-bit MRO-PUF achieves good resistance less than 0.55 prediction accuracy. These results show that the proposed MRO-PUF with longer challenge bits has strong resistance against various non-linear machine learning attacks.

The required time for machine learning attacks are summarized in Table 1. The experiments are run on a Linux PC with Intel Xeon E5-2630 v2 2.60 GHz CPU. The number of CRPs used for training is 10^4.

4.5 Response Generation Time

The required time to generate a response of the proposed MRO-PUF is T_{meas}, which is equal to $N_{\text{meas}}T_{\text{CLK}}$ for 2-MRO-PUF. For the 16-bit 2-MRO-PUF with $N_{\text{meas}} = 50$, it is

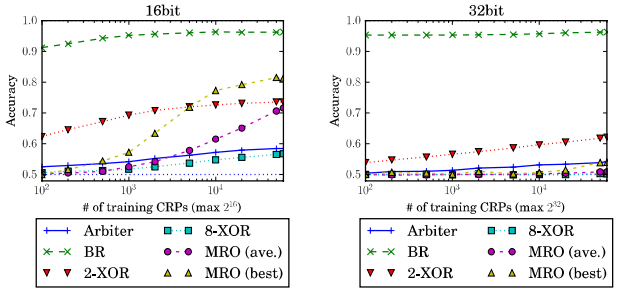

Fig. 16 Resistance against machine learning attack by random forest.

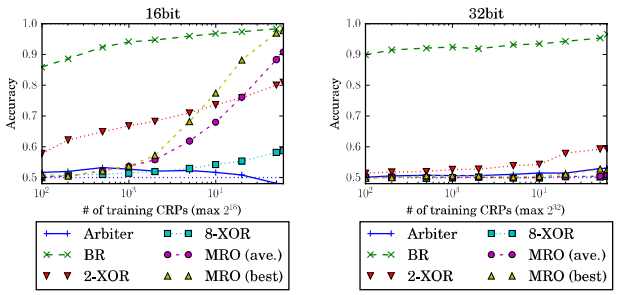

Fig. 17 Resistance against machine learning attack by boosting.

Table 1 Time required for machine learning attacks [s].

	16 bit				32 bit			
	SVM	ES	RF	Bst.	SVM	ES	RF	Bst.
Arbiter	0.06	25.3	0.69	2.04	0.23	39.2	1.16	0.05
BR	0.03	12.7	0.62	1.87	0.15	17.3	0.76	0.03
2-XOR	0.27	23.6	0.69	2.02	0.60	38.6	1.15	0.05
8-XOR	0.26	24.2	0.69	2.02	0.63	41.7	1.16	0.05
MRO	0.28	14.8	0.69	2.02	0.62	19.5	1.15	0.05

about $T_{\text{meas}} = 50 \times 2.62\,\text{ns} = 131\,\text{ns}$. This is longer than that of the arbiter PUF having the same challenge bits, which is about 1.2 ns, and comparable to that of the BR-PUF, which is about 100 ns. Although relative throughput of the proposed PUF is inferior to those of arbiter-based PUFs, it can still be considered sufficiently fast and acceptable in most of the practical applications.

5. Related Works

The proposed MRO-PUF is based on two key ideas: modulo operation to enhance resistance against machine learning attacks and route-selectable RO structure for area efficiency. This section reviews existing PUFs utilizing the similar ideas, and compares them with the proposed MRO-PUF.

First, for the modulo operation, an RG-DTM PUF [23] and a DC-ROPUF [24] are based on the similar concept.

The RG-DTM PUF [23] is an extension of the arbiter PUF. It divides time-difference of two signal routes into multiple regions. By alternatively assigning 0 or 1 to the divided regions, the RG-DTM PUF generates responses that are difficult to predict. This region-dividing technique is very effective to improve resistance against machine learning attacks; the modulo operation in the MRO-PUF can be

considered fundamentally the same as the region division in the RG-DTM PUF, but it has a weakness in its robustness. Since the response of the RG-DTM PUF is determined by the "difference" of the delay time, it is easily affected by the fluctuation of the temperature and the supply voltage. On the contrary, the MRO-PUF determines its response by the "ratio" of the delay times of two ROs by Eq. (18), which can cancel the delay fluctuation that appears similarly in the two ROs. This means that the proposed MRO-PUF has an advantage to the RG-TDM PUF in terms of its robustness. In addition, the circuit structure of the RG-DTM PUF depends on the number of the divided regions, which affect the performance of the resistance against machine learning attacks. On the other hand, the MRO-PUF can adjust such performance just by changing the value of N_{meas} without modifying its circuit. This flexibility is another advantage of the proposed MRO-PUF.

The DC-ROPUF [24] utilizes two ROs and its response is an instantaneous output value of one of the two ROs at a timing when the other RO counts up a certain value. Although this idea is very similar to our 2-MRO-PUF, the DC-ROPUF has a problem for its area efficiency. The basic structure of the DC-ROPUF is the conventional RO-PUF [11], which is known for its inefficiency of the circuit area since it requires ROs exponential to the number of challenge bits. For example, to realize a 16-bit PUF, the DC-ROPUF requires $7 \times 2 \times 256 = 3584$ inverters, whereas the 2-MRO-PUF only requires $17 \times 2 \times 2 = 68$. In addition, the DC-ROPUF also requires large 16-bit multiplexers to select ROs according to the challenge bits. Therefore, our MRO-PUF is more area-efficient than the DC-ROPUF.

Second, for the route-selectable structure, the aforementioned DC-ROPUF [24] and a selective RO-PUF [25] are the ones that utilize route-selectable ROs to enhance area efficiency. However, as described above, the DC-ROPUF does not adopt fully route-selectable ROs but also has area-consuming conventional RO-PUF structure where multiple ROs are arranged in parallel. As a result, the DC-ROPUF is less area-efficient than the proposed MRO-PUF. The DC-ROPUF also has the problem for physical layout, which is originally the problem of the conventional RO-PUF; all the ROs should have identical layout with a regular placement. Since the proposed 2-MRO-PUF uses only two ROs, its layout is easier than the PUFs based on the RO-PUF. The selective RO-PUF [25] resolves the above problems of area efficiency and layout difficulty by adopting a fully route-selectable structure. However, the selective RO-PUF is vulnerabile to machine learning attacks since its response is determined by calibrated frequency difference of two ROs through linear calculations, which can be easily modeled by a linear classifier such as SVM.

In summary, a part of the techniques used in the MRO-PUF has been already proposed in the existing works, our MRO-PUF is only the work that can enhance resistance against machine learning attacks, robustness, and area-efficiency simultaneously.

6. Conclusion

In this paper, we proposed a new MRO-PUF that utilizes an instantaneous output of a ring oscillator as a response. Unlike the existing PUFs whose CRPs are linearly separable, a response of the MRO-PUF is determined by a non-linear and discontinuous modulo operation, which makes its CRPs difficult to be predicted by a linear classifier, such as SVM. We further improved the robustness of the MRO-PUF by introducing an additional ring oscillator to compensate for temperature and supply voltage fluctuations. Through the experiments by SPICE and software simulations, it is shown that the proposed MRO-PUF achieves 15 times stronger resistance against machine learning attacks compared to the existing arbiter PUF and the BR-PUF, while keeping the sufficient level of the basic performances, i.e., the uniqueness and robustness.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant No. 26280014 and 17H01713. The authors also acknowledge support from VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] B. Gassend, D. Clarke, M. van Dijky, and S. Devadas, "Silicon physical random functions," Proc. Computer and Communication Security Conference, pp.148–160, ACM, 2002.
[2] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, and S. Devadas, "PUF modeling attacks on simulated and silicon data," IEEE Trans. Inf. Forensics Security, vol.8, pp.1876–1891, Nov. 2013.
[3] F. Ganji, S. Tajik, F. Fäbler, and J. Seifert, "Strong machine learning attack against PUFs with no mathematical model," Proc. Cryptographic Hardware and Embedded Systems, pp.391–411, Springer Berlin Heidenberg, June 2016.
[4] A. Vijayakumar, V.C. Patil, and C.B. Prado, "Machine learning resistant strong PUF: Possible or a pipe dream?," Proc. International Symposium on Hardware Oriented Security and Trust, pp.19–24, IEEE, May 2016.
[5] J.W. Lee, D. Lim, B. Gassend, G.E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," Digest of Technical Papers of Symposium on VLSI Circuits, pp.176–179, IEEE, 2004.
[6] U. Rührmair, S. Devadas, and F. Koushanfar, "Security Based on Physical Unclonability and Disorder," in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang, eds., ch. 4, pp.65–102, Springer New York, Aug. 2011.
[7] B. Gassend, "Physical random functions," Master's Thesis, 2003.
[8] J. Kelsey, B. Schneier, D. Wagner, and D. Hall, "Side channel cryptanalysis of product ciphers," Proc. European Symposium on Research in Computer Security, pp.97–110, Sept. 1998.
[9] P.C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," Proc. Annual International Cryptology Conference, pp.104–113, Berlin, Heidelberg, Springer Berlin Heidelberg, 1996.
[10] J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," Proc. Cryptographic Hardware

and Embedded Systems, pp.63–80, Sept. 2007.
[11] G.E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," Proc. Design Automation Conference, pp.9–14, 2007.
[12] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Rührmair, "The bistable ring PUF: A new architecture for strong physical unclonable functions," Proc. International Symposium on Hardware Oriented Security and Trust, pp.131–141, IEEE, May 2011.
[13] X. Xu, U. Rührmair, D. Holcomb, and W. Burleson, "Security evaluation and enhancement of bistable ring PUFs," Proc. International Workshop on Radio Frequency Identification, pp.3–16, June 2015.
[14] G. Hospodar, R. Maes, and I. Verbauwhede, "Machine learning attacks on 65nm arbiter PUFs: Accurate modeling poses strict bounds on usability," Proc. Workshop on Information Forensics and Security, pp.37–42, Dec. 2012.
[15] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, 2006.
[16] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," Proc. International Conference on Reconfigurable Computing, pp.298–303, Dec. 2010.
[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," Journal of Machine Learning Research, vol.12, pp.2825–2830, 2011.
[18] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," Proc. ACM Conference on Computer and Communications Security, pp.237–249, 2010.
[19] H.G. Beyer and H.P. Schwefel, "Evolution strategies – A comprehensive introduction," Natural Computing, vol.1, no.1, pp.3–52, 2002.
[20] V. Svetnik, A. Liaw, C. Tong, J.C. Culberson, R.P. Sheridan, and B.P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," J. Chem. Inf. Comput. Sci., vol.43, no.6, pp.1947–1958, 2003.
[21] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol.55, no.1, pp.119–139, 1997.
[22] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," Journal of Machine Learning Research, vol.11, pp.743–746, 2010.
[23] M. Shiozaki, K. Ogawa, K. Furuhashi, T. Murayama, M. Yoshikawa, and T. Fujino, "Security evaluation of RG-DTM PUF using machine learning attacks," IEICE Trans. Fundamentals, vol.E97-A, no.1, pp.275–283, Jan. 2014.
[24] M. Ikeda, H. Kang, and K. Iwamura, "Direct challenge ring oscillator PUF (DC-ROPUF) with novel response selection," Proc. Global Conference on Consumer Electronics, Oct. 2017.
[25] M. Yoshikawa, T. Asai, M. Shiozaki, and T. Fujino, "Selective ring oscillator PUF with statistics correction technique and its evaluation," Trans. Institute of Systems, Control and Information Engineers, vol.25, no.1, pp.1–9, 2012 (in Japanese).

Masayuki Hiromoto received B.E. degree in Electrical and Electronic Engineering and M.Sc. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University in 2006, 2007, and 2009 respectively. He was a JSPS research fellow from 2009 to 2010, and with Panasonic Corp. from 2010 to 2013. In 2013, he joined the Graduate School of Informatics, Kyoto University, where he is currently a senior lecturer. His research interests include VLSI design methodology, image processing, and pattern recognition. He is a member of IEICE and IPSJ.

Motoki Yoshinaga received B.E. degree in Electrical and Electronic Engineering and M.E. degree in Communications and Computer Engineering from Kyoto University in 2015 and 2017 respectively. He is currently with Canon Inc., where he is involved in developing hardware accelerators for image recognition. He is a member of IEEE.

Takashi Sato received B.E. and M.E. degrees from Waseda University, Tokyo, Japan, and a Ph.D. degree from Kyoto University, Kyoto, Japan. He was with Hitachi, Ltd., Tokyo, Japan, from 1991 to 2003, with Renesas Technology Corp., Tokyo, Japan, from 2003 to 2006, and with the Tokyo Institute of Technology, Yokohama, Japan. In 2009, he joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently a professor. He was a visiting industrial fellow at the University of California, Berkeley, from 1998 to 1999. His research interests include CAD for nanometer-scale LSI design, fabrication- and reliability-aware design methodology, and performance optimization for variation tolerance. Dr. Sato is a member of the IEEE and the Institute of Electronics, Information and Communication Engineers (IEICE). He received the Beatrice Winner Award at ISSCC 2000 and the Best Paper Award at ISQED 2003.